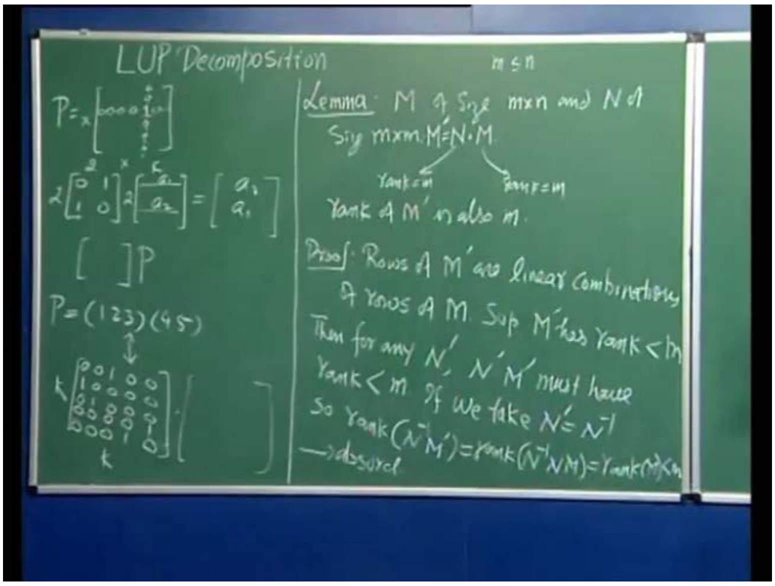
Computer Algorithms - 2
Prof. Dr. Shashank. K. Mehta
Department of Computer Science and Engineering
Indian Institute of Technology, Kanpur

Lecture - 13
Matrix Decomposition

Hello, today we will discuss LUP decomposition of matrices. This decomposition will break a matrix into product of three matrices. This will be a square matrix, which is a lower triangular matrix, which means all the entries above the triangle that is above the diagonal will be 0. This will be a matrix, which will have all the entries below the diagonal to be 0 and P will be a permutation matrix.

(Refer Slide Time: 00:57)



So, let us just first of all make sure we know what a permutation matrix is P is a square matrix of size x by x. It is a permutation matrix, if there is exactly 1 1 in each row and each column and all other entries are 0, so any given row and column has only 1 1. So, identity matrix is a permutation matrix. Similarly, in 2 by 2 this is a permutation matrix. The application of a permutation matrix on another matrix from left, so suppose this is this is 2 by 2 matrix. Let us say this is 2 by k matrix, then the application of this matrix on this, is effectively permuting the rows of this matrix. Similarly, if I am applying a permutation matrix from the right hand side by multiplying from right. We are permuting the columns of this matrix.

What exactly is happening in this case, is that we have over here row a 1 and here row a 2, when I multiply, I am going to get the entries of a 2 first. Because I multiply here, I will get the entries of a 1. So, this is exchanging the row here in 2 by 2. You do not have too many other cases identity or this. Similarly, if it is a larger matrix it will be some general permutation. Now, in general a permutation can also be expressed in form of cycles. So for example, if I have 1, 2, 3, 4 and 5, this means that 1 goes to 2, 2 goes to 3, 3 goes to 1, 4 goes to 1 and 5 goes to 4.

So, the corresponding matrix, in the sense that the same effect it causes on the object to it is right, can be achieved by noticing that this has to be a 5 by 5 matrix. The first row, if I am multiplying from left, so I am permuting the row. So, the first row must go to the second position. To do that, what we have to do is put 1 in the second place 0 0 0 0 . The second row must go to third place, so this will be 0 1 0 0 0 and the third row should go to the first position, so it will be 0 0 1 0 0. Notice that these 3 rows are going to effectively achieve this permutation. The fourth should go to the fifth place and the fifth to the fourth.

So, to fourth and fifth, those two we have. So, it is easy to consider such a permutation from this and similarly 1 can go back. Further, if I have to multiply 2 permutation matrices, we can actually directly perform that through this representation. Such a computation can be done in the time of the individual dimension. So, if this is k by k, multiplication of 2 k by k matrices is certainly. Then k square, but this we can do in order k times. Now, this information will be coming in handy later. This 1 result that I would like to describe here before we proceed, do discuss LUP decomposition and that is the following.

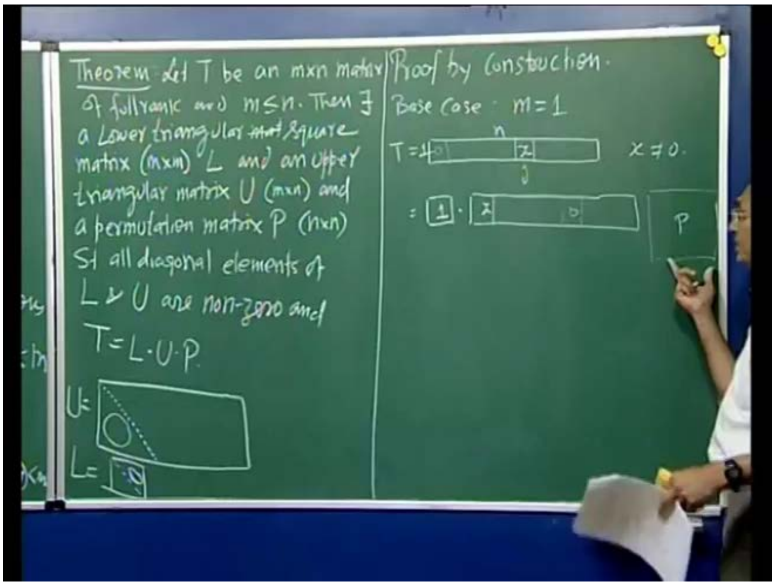
Suppose, we have a matrix M of size m cross m and we have N of size m cross m and we consider the product N dot M. Suppose, and further we assume that m is less than equal to n, for convenience assume that the matrix M is full range. What does that mean? It means that were m is less than equal to n, there is a sub matrix of size m cross m which is non single. That information is equivalent to the fact that in this m cross n matrix the rows are linearly independent. Equivalently, in terms of columns there exist m column which are linearly independent in which n we may not know. Now, suppose we have a matrix n which is non singular, hence this also has rank equal to m. In that case, the product is M prime.

Then, the claim is that the rank of M prime is also M. To prove this, observe that the rows of M prime are combinations of the rows of M. So, the rows of M prime are linear combinations of rows of M, why because, if you take look at them, you are multiplying, say you take the first row of N and you multiply. So, those coefficients are multiplied to various rows of M and you combine them. That makes the first row of M prime. So, each of these each of the row vector of M prime is combination of rows of M. Now, suppose M prime has rank strictly less than k and less than M, then for any N prime, M prime must have rank strictly less than M.

The reason is that once again if the rows of this are linearly dependent, then these are further combination of the same rows. They cannot be linearly independent. Hence, this must also have a rank strictly less than m. Now, if we take N prime to be N inverse, notice that our N is a full rank matrix. Hence, the inverse exists full rank for a square matrix. This is equivalent to non singular. So, M prime is rank of N inverse and N M is equal to rank of M, which is less than m.

That is absurd. Hence, we conclude that if this is full range, note that the rank cannot be more than the smaller of the two dimensions. The smaller is M, so this is M and this is N that the rank of M prime must also be same. So, this is a small result that we are going to need. Now, let us talk about LUP decomposition. What we will show is going to be stated in a theorem.

(Refer Slide Time: 11:33)



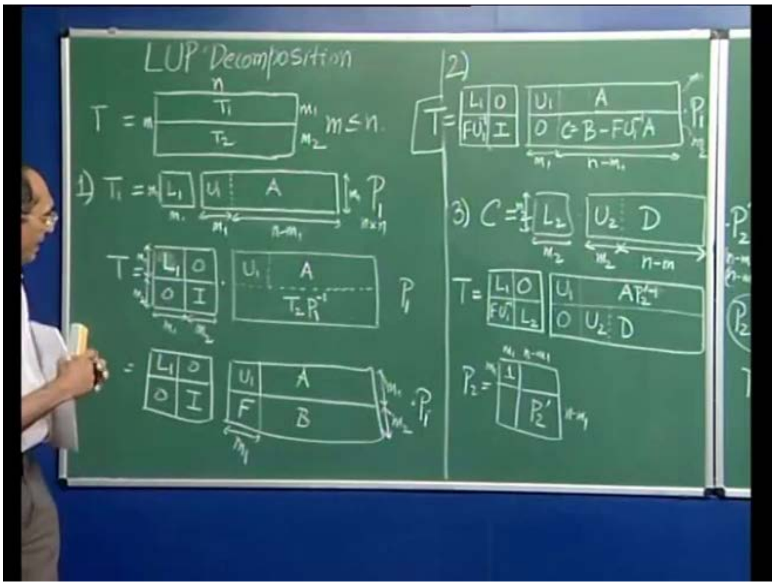
Let me state that, say let T be an m cross n matrix of full rank and m is less than equal to n. Then, there exists a lower triangular matrix, square matrix which is actually m cross m, that is L and an upper triangular matrix U that will be of size m cross n and a permutation matrix P of size n cross n, such that all diagonal elements of L and U are non-zero and our original matrix is the product of L U and P. Now, 1 word about this matrix is this matrix is not a square matrix. So, what we mean by upper triangle is that in the diagonal these are the elements A i i.

So, everything below this is 0. This is the way U should look like. Of course, L is this. It would be 0. So, we will prove this theorem by actually giving an algorithm to construct the 3 matrices. Now, the base case will be when m is equal to 1. So, let us say, proof by construction. We will use recursion in construction and our base case is when m is equal to 1. So, we have our T which looks like a 1 by n matrix. Now, all entries of this cannot be 0, because it is of rank 1. There has to be some place, some j th location, say j location where we have x and x is non zero.

So, one can rewrite this in the fashion 1 times x here, in case j is 1. In that case, we do not have to worry too much. But, say this is 0 and then we will have the value. Whatever, this value is a 0, this is 0 here and a permutation matrix P which swaps the first position with the j th position, we know how to write such a permutation matrix and leave everybody else in it is own position. This is a lower diagonal matrix, lower triangular matrix. The reason is that it has only 1 entry and that falls on the diagonal.

As we require all the diagonal entries must not be 0, and indeed it is non zero. In the second matrix, the only diagonal entry is 1 in the position 1 1. That, by construction is non zero. There is no entry below this diagonal. Hence, we do not have to worry about this. This will qualify as an upper triangular matrix. This is a permutation matrix. Hence, we satisfy the requirement for the base case. Now, that we are done with the base case, let us take a arbitrary matrix of size m cross n.

(Refer Slide Time: 17:25)



Once again, remember that m is less than or equal to n. We split this matrix anywhere. So, this is some m 1 and this is m 2. Let us call this sub matrix T 1, and this is T 2. Now, notice that since m is less than equal to n, the full rank means all the row vectors are linearly independent. As a result, the row vectors in T 1 are linearly independent. So they are in T 2. Hence, T 1 and T 2, themselves are of a full rank. We are in position to perform LUP decomposition on them. So, let us suppose in step 1, we do LUP decomposition by recursive call of this algorithm and we get the lower and upper triangular matrices.

So, I am going to in this. I am going to call this as U 1 and the rest of the sub matrix as A. Then, we have a P matrix. Note that, this is m 1 by m 1. This is m 1 by n. Maybe, I will split this as m 1 and this is n minus m 1. This is of course, n cross n permutation matrix. This is m 1. Now, that we have this decomposition, we can rewrite T as T 1 0 0 identity. This is m 1 and this is m 2 by this m 2, this is m 1 and this is m 2. This is U 1 and this is A. We have T 2 here and we have a P there. Now, note that the upper part is unaffected, because when you multiply, the 0 does not contribute anything. The 0 multiplies to T 2.

It does not change anything and we really get T 1, U 1, T 1, A and then, of course P. On the other hand, this portion is also unaffected I times T 2 is T 2, but after that this is going to be affected, because of P. To deny that change, we will have to multiply by inverse of P. Note, that every permutation matrix is full rank, is non singular and its inverse exists. Now, this is L 1 and so this L 1, is a starting decomposition of T. But, we have yet to go further. The reason is, this is indeed a lower triangular matrix. Because, this is a lower triangular, this is identity matrix which is lower and upper triangular and is 0, but this is not. Now we have to further modify.

So, in the next step, maybe we will just re-label this in the following fashion. We will call this T I 0 0 here. Let us call U 1. This portion is A and this is B. Let us call this F. So this is still m1. So, we have m1 and m 2 here and of course multiply to P. Now, in the next step we will attempt to make this part 0. Now, to do that we have to ensure that the contribution, which is F when we go in this part is still maintained. For that we will set this to 0 and this will be such that this when multiplied here gives me F. So, that can be done by replacing this by F U 1 inverse. This will stay there and we have A. This is say U 1. This is A and this we will be set to 0.

We will have a P here. Let us look at the impact of this change. This, when multiplied to this column, this part of the matrix, we are going to get F back. That contribution will be same time, but when we multiply this here, this was B 1 here. So, B will be still there. We will also get F U inverse A, which we must eliminate. So, we got to replace this by B minus F U 1 inverse A. That will make things equal to the original product. Now, let us call this matrix C. This is still equal to T. Now, we knew from our theorem and from induction hypothesis, that the all the entries of the diagonal entries of L 1 are non-zero. All the diagonal entries of U 1 are non zero.

So, first thing is since diagonal entries of U 1 are non zero and this is a upper triangular matrix, from our last lecture we know that such a matrix has inverse. We can compute that in the same time as matrix multiplication of size m 1 by m 2. Hence, this step is valid computation of U 1 inverse. Secondly, since all these entries are non-zero, the diagonal entries are non zero. These are also non zero. Hence, this matrix over here with F U 1 inverse, this is a lower triangular matrix. Once again, in principle its inverse exists. Now, suppose this, we call some L prime, then L prime inverse T is this matrix times P. At the start, I had pointed out that if this is a full rank matrix, then this product is also a full rank matrix.

Hence, this product is a full rank matrix, but this is a permutation matrix. It just permutes the column. It does not change the rank. Hence, this matrix must be a full rank matrix. Once again, because this is full rank, this part should be all linearly independent rows. So, this sub matrix is also a full rank matrix which actually is m 2. Notice that, this is of size m 1 and this is size m 2. So, this is a full rank matrix, but these entries are all 0. Hence, this sub matrix must also be full rank. Once again, we have a full rank matrix. So, we can recursively apply this LUP decomposition on C. So, in our step 3, we compute C. We have lower triangular matrix.

So, L 2 U 2 and some matrix D followed by P 2 prime. Choose to write P 2 prime. So, let us try to first verify our dimensions and all this was m 2 by n minus m 1 size. So, both of the dimensions of L 2 are m two. This m 2 by n minus m 1 size, this is m 2. This we will split it here. This is m 2. This is n minus m 1 minus m 2, but m 1 plus m 2 is n. So, this is n minus m. This should be a square matrix of size n minus m 1. So, this is n minus m 1 cross n minus m 1, because this is going to be multiplied to this matrix. Now, again keep in mind that the diagonal entries are non zero. Here, diagonal entries are 0.

Here, so let us try to rewrite our T recognizing this factorization. So, we will put L 1 here, F U 1 here and we will put L 2 here. This will be 0. You have U 1 and 0 here. Let us see, if we put P here, we will put a P 2 and a P 1, where P 2 will be P 2 prime here, A unit matrix over here, of size m 1 by m 1. So, this is m 1. This is n minus m 1. This is also n minus m 1. Now, is this correct? So, let us first look at this. This is an expression. This was P 1. I would like to make this P 1, because we have a P 2 sitting here. The effect of multiplying the upper part with L 1 will be same, except that we have now P 2 sitting in between to deny that effect.

We will put A P2 inverse. Now, if you just look at the multiplication of this part, it is like L 1 multiplied to U 1, L 1 multiplied to A P2 inverse. The reason we do not have to multiply anything here. It is P 2 prime inverse P 2 prime, a smaller matrix. The reason is that in P 2, this portion is identity which does not affect this one and P 2 prime is going to be multiplied to A. So, that is cancelled, the lower part when we go about multiplying this way, the contribution from identity was that we were getting directly C followed by multiplication of C 1. This time we are going to multiply L 2. This is going to be 0, but L 2 has to be U 2 D. This full vector is sitting here.

Now, L 2 is being multiplied to this part followed by P 2 but since this corner is identity, we are going to be effectively multiplying identity to this and P 2 prime to this. So, we will exactly get C and followed by P 1. So, this is now the new form of T. At this stage what we have is a clear lower triangular matrix. Because, here this is lower triangular all entries are non zero at the diagonal. Here, all entries here are non zero at the diagonal, so the entire diagonal is non zero. So, this is what we want as a lower triangular matrix. In this part, here it is an upper triangular matrix. Exactly the same is true.

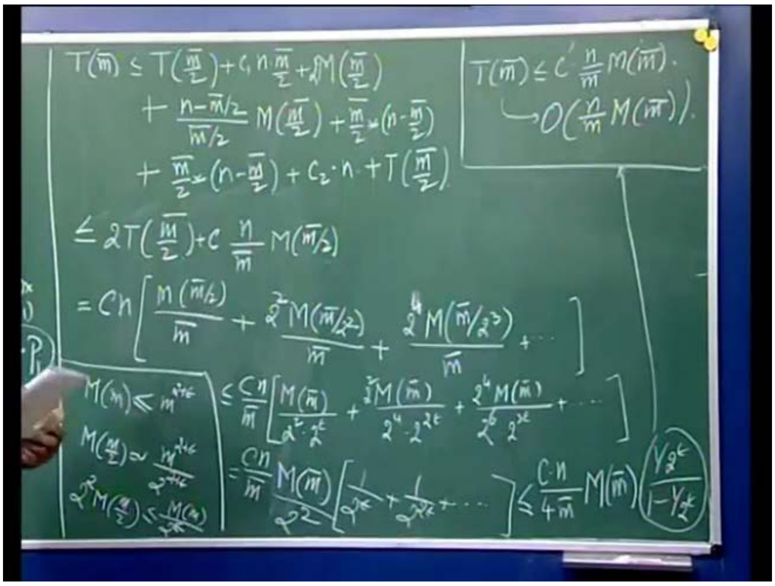
All the diagonal entries are non zero, only thing is the zeroes are on this side. So this is 0 here. It is 0 here. It is 0. So, this matrix is actually a square upper triangular matrix. This is also upper triangular matrix. Hence, we have completed the transformation of the matrix T into product of these 3, where these are 4 matrices. Now, we can replace this by a product matrix P, which is the product of 2 permutation matrices. It is also a permutation matrix. So, we get a complete decomposition into lower triangular and upper triangular, a permutation. Now, at this stage we will like to determine the cost. Now, let us take each step and count the cost of each computation in the first step.

We have taken the upper half matrix. We are going to pay the price of transforming and m 1 by n size matrix into this product. Now, to do all this, first thing I am going to assume that T of m denotes the cost of transforming an m by n matrix into loop P matrices. I will not indicate n here, because this is a background parameter which will not show up to often the cost of this computation. Therefore, it will be T m 1. Now, let us say m bar denotes the nearest power of 2 greater than or equal to m. We always break as close to middle as possible, that is if this is even than we will just split in the middle. If it is odd then one will be slightly off.

In that case both m 1 and m 2 will be less than equal to m bar by 2. Then, m 2 will be less than equal to m bar 2. This will always be the case. So, we will simplify our life and think of m as m bar. This will be less than equal to first factor, T m bar by 2 that is coming from this step. In the next step, well in the very same step, what we have done is we have multiplied P 1 inverse to T 2. Now multiplication by a permutation matrix, so remember P inverse is also permutation.

This is like permuting the columns, but permuting the columns we do not have to do a regular matrix multiplication. We can simply read the permutation and write down the column. Therefore, this can be done in order of the size of this matrix itself and that would be n into m bar 2. So, I will add a cost, some c 1 time n into m bar by 2. That would be the cost of this step.

(Refer Slide Time: 38:06)



Then, we have computed inverse of U 1 multiplied it to n, then multiplied it to A and then you subtracted this from this. So, there are 3 steps. Now, computation of U 1 inverse. We have already shown in the last lecture that the inverse cost is of the order of the same as that of multiplication and this is true for triangular matrices. So, this I will denote by writing down M m bar by 2, where M is the cost of multiplying 2 matrices of this size that is the cost of inverse. Now, we multiply this to another matrix of same size, so which is again the same thing. So, we can say put 2 here. Then, we are multiplying this to a larger matrix in general. A is M by 2 by n minus M by two.

Now, notice that if you have to multiply x by x matrix to x by y matrix, y is say larger than x, then we can think of this as split in 2 matrices of size x by x and x by x, etcetera. So, that cross can be M x into y over x, because that many sub matrices are sitting. So, that many multiplication. So, we can add this. We have a cost of n minus m bar by 2 of M m bar by 2. That accounts for this multiplication and then subtraction is 1 unit work for each entry. So, that costs us m bar by 2 into n minus m bar by 2. That many entries are there in this matrix. So, next step is the application of P 2 prime inverse to A. As we have seen earlier, we can do this by simply permuting the column which will cost us as much as the size of this matrix.

That would be m bar by 2 into n minus m bar over 2. The only thing which looks threatening is the multiplication of these 2 matrices. Both of them are n cross n. If I try to write m n that would be very costly, but recall that at the start we said these matrices need not be kept as matrices. We can store them in the form of product of cycle and the multiplication of these to permutation can be done in order n rather than order m n. So, this whole thing will cost us some c 2 times n that is the product of the 2 m cross n matrix. There is 1 more operation we have performed. This is where we have applied the LUP decomposition on a C.

C is of size m 2 by n minus m 1. Since, we have kept n as a suppressed parameter, we will not worry. We will think of this as n which does not really heard and this will be M by 2. So, this decomposition also cost as T M by 2. Now, that accounts for all the computations we have performed. Now, let us try to clean this up. We have 2 T m bar by 2, because of these two. If you account for this and this together this is going to be n by m bar by 2. So, 2 of that M. So, we can actually put some constant here. Other than this, we will put some C M m bar by 2 that accounts for this entry and this entry together.

Notice that, this multiplication is at least square of this. So, this entry is more than m bar times n into some constant. Hence, this entry is subsumed by this, this entry is subsumed by this, this is subsumed by this and this is also subsumed by this. We do not have to put any more entry. For a suitable choice of constant this can be taken as an upper bound to T m bar. Now, all we have to do is simplify this and solve for the value of T m prime. So, if I expand this I am going to get this as C n M m bar by 2 by m bar plus 2 of the lower size. So, there will be 2 here. This will be replaced by m bar by 2 square. This will be m bar by 2.

This will be 2 square here. One more term, may be this will be twice of this. So, we have a 2 cube here, M m bar by 2 cube over m bar by 2. So, that makes it 4. This is the expansion of this recurrence relation. Now, recall that M of m is about 7 to the power 2 plus epsilon. So, we can say that M m bar by 2 is approximately M plus 2 epsilon by 2 power 2 plus h or we can say that there exists an epsilon from which we can bind this.

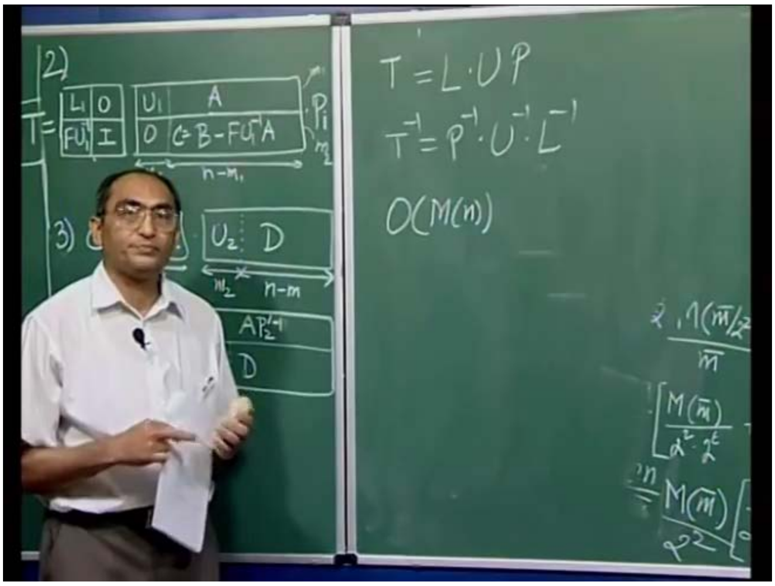
So, if I take 2 to the power 2 here, we can say this is bounded by M n over 2 power epsilon. So, we will use this inequality and rewrite this as C n by m bar.

That makes it M of m bar 2 is 2 to the power 2 into 2 to the power epsilon plus M m bar. We have a 2 power 2 here. We have 2 power 4 here and 2 power 2 epsilon, 2 power 4 M m bar, 2 power 6 and 2 power 3 epsilon. Now, we combine these and get some constant times n over m bar. We can take M m bar out. So, this will be M of m bar. Notice that, powers of 2 are also cancelled and they all are 2 power 2 at the bottom. So, we can leave that here and we have 1 over 2 epsilon plus 1 over 2 power 2 epsilon. So, this is less than equal to 1 over 2 power epsilon divided by 1 minus 2 over 2 epsilon.

So, we have C times n over 4 m bar M of m bar times. This is a constant 1 divided by 2 power epsilon over 1 minus 1 over 2 power epsilon. This is the state constant and the entire cost is T of n bar is less than equal to some constant prime times n over m bar M of m bar or this is of the order of n by m M m bar. Now, indeed if the starting matrix was square, then this will cancel out and we will find that the total cost of LUP decomposition is same as the cost of multiplication of 2 square matrices. Notice that, m bar can be at most 2 times M and that is only going to be a constant factor.

We can think of M of m bar as some constant times M of m. So, we have finally, the computation of inverse of a triangular matrix, computation of matrix multiplication of matrices and computation of decomposition into triangular matrices. All of them have the same cost. Now, the last thing I would like to point out is that, maybe we can just wipe this out that if you have an arbitrary matrix, the square matrix which is non-singular, then how do we compute the inverse and what would it cost?

(Refer Slide Time: 51:53)



So, suppose we start with the square matrix T, I can then convert this into LUP. Note that, all of them will be square now. All them will be non-singular and these two will be lower triangular and upper triangular, for which we have an algorithm to compute the inverse. So T inverse will be P inverse U inverse L inverse. So, that involves computation of inverses. This is trivial, then the product that will give you, since each step costs order M of the size of the matrix. Therefore, this entire process will also cost us order M n times. So, this is where we close the discussion of the matrices.